wight Documentation

Release 0.3.3

Bernardo Heynemann

September 09, 2013

CONTENTS

Wight is a continuous load testing tool.

It enables developers to keep track of whether their code is evolving performance-wise.

It’s a great tool to spot performance issues and bottlenecks.

CHAPTER

ONE

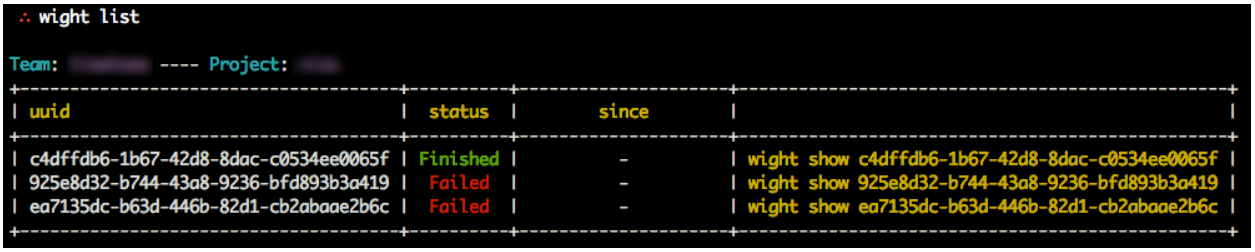
COMMAND LINE INTERFACE - CLI

After installing Wight, you'll have a few commands available to you. One of them is the *wight* CLI.

The CLI is the toolbelt that allows developers to interact with wight when scheduling tests or adding new projects.

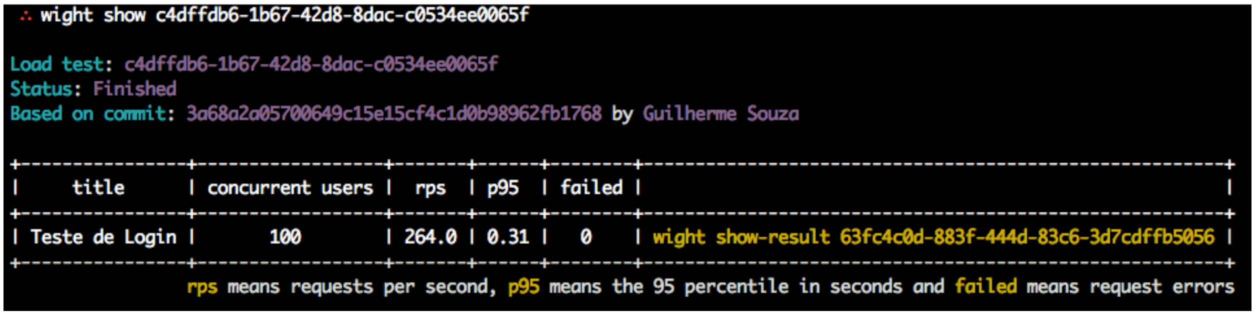
The most important thing is the endpoint that the CLI will be talking to. We call that the *target*.

To learn more about how to use the CLI, head to the *Getting Started* page.



```
 ∴ wight list

Team: [redacted] ---- Project: [redacted]
+--------------------------------------+----------+--------------------+-------------------------------------------------+
| uuid                                 |  status  |       since        |                                                 |
+--------------------------------------+----------+--------------------+-------------------------------------------------+
| c4dffdb6-1b67-42d8-8dac-c0534ee0065f | Finished |         -          | wight show c4dffdb6-1b67-42d8-8dac-c0534ee0065f |
| 925e8d32-b744-43a8-9236-bfd893b3a419 |  Failed  |         -          | wight show 925e8d32-b744-43a8-9236-bfd893b3a419 |
| ea7135dc-b63d-446b-82d1-cb2abaae2b6c |  Failed  |         -          | wight show ea7135dc-b63d-446b-82d1-cb2abaae2b6c |
+--------------------------------------+----------+--------------------+-------------------------------------------------+
```



```
 ∴ wight show c4dffdb6-1b67-42d8-8dac-c0534ee0065f

Load test: c4dffdb6-1b67-42d8-8dac-c0534ee0065f
Status: Finished
Based on commit: 3a68a2a05700649c15e15cf4c1d0b98962fb1768 by Guilherme Souza

+----------------+------------------+-------+------+--------+--------------------------------------------------------+
|     title      | concurrent users |  rps  | p95  | failed |                                                        |
+----------------+------------------+-------+------+--------+--------------------------------------------------------+
| Teste de Login |       100        | 264.0 | 0.31 |   0    | wight show-result 63fc4c0d-883f-444d-83c6-3d7cdffb5056 |
+----------------+------------------+-------+------+--------+--------------------------------------------------------+
                  rps means requests per second, p95 means the 95 percentile in seconds and failed means request errors
```

CHAPTER

TWO

WEB REPORTS

While the way of using Wight daily is through the CLI, it comes bundled with reports that make it easy to understand more about your application.

Wight breaks the results into three different units: *tests*, *pages* and *requests*.

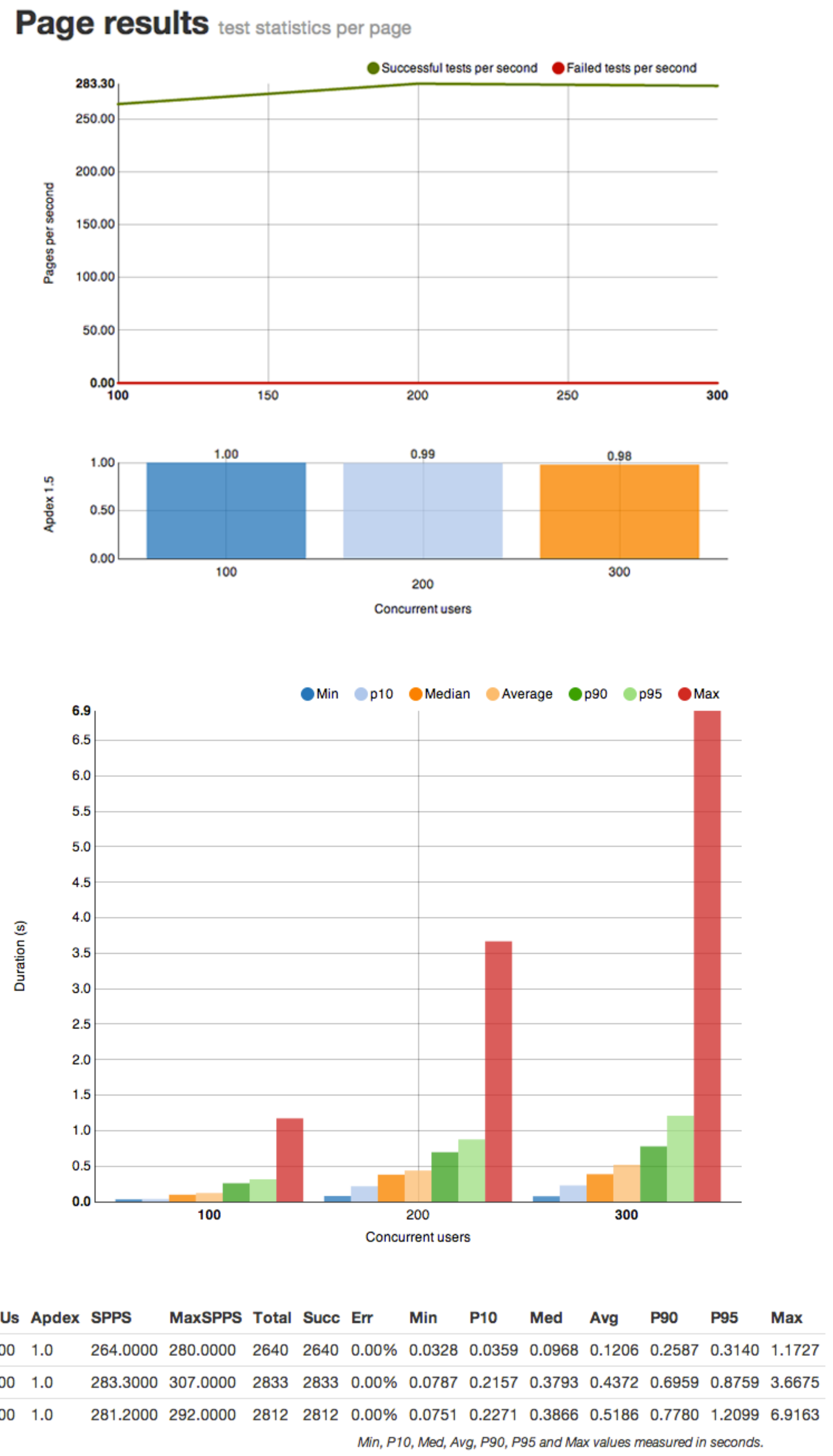
Tests are the actual units you write using funkload. Wight runs your tests many times and report them back to you.

Each test requests pages to the product being tested. Each page requested gets tracked independently. In a given test you may request more than one page (get and post a login page, then get an index page, for example).

Requests only differ from the pages if the test being run is not being run under simple-fetch. Simple-fetch only gets the page HTML, but does not parse it. If the test is not run with simple-fetch, then the page is parsed and all the resources (js, css, images) are also downloaded.

This means that usually the *# of requests >= # of pages >= # of tests*.

An example of the reports for a test (just a sample portion of the report data shown for clarity):



Page results test statistics per page

CUs	Apdex	SPPS	MaxSPPS	Total	Succ	Err	Min	P10	Med	Avg	P90	P95	Max
100	1.0	264.0000	280.0000	2640	2640	0.00%	0.0328	0.0359	0.0968	0.1206	0.2587	0.3140	1.1727
200	1.0	283.3000	307.0000	2833	2833	0.00%	0.0787	0.2157	0.3793	0.4372	0.6959	0.8759	3.6675
300	1.0	281.2000	292.0000	2812	2812	0.00%	0.0751	0.2271	0.3866	0.5186	0.7780	1.2099	6.9163

Min, P10, Med, Avg, P90, P95 and Max values measured in seconds.

CHAPTER

THREE

API

To support the CLI, Wight comes baked in with a fully-fledged API, meaning that if you want to automate it using something other than the CLI, it's very simple.

The API is "RESTish", so it's fairly easy to grasp. Sometimes, though, for the sake of performance or clarity, the team chose to create routes that improve the whole system in spite of purity.

To learn more about the API, head to the *API* page.

CHAPTER

FOUR

CONTENTS

4.1 Getting Started

TBW.

4.2 Web Reports

TBW.

4.3 CLI Usage

Running the Wight CLI is as simple as:

```
$ wight --help
```

This will show you the full list of commands (a summary of this page).

To run a wight command, like *list* for example:

```
$ wight list
```

4.3.1 Target Commands

The target is the endpoint the CLI will use to connect to wight.

target-set

Sets target for wight to use. It requires an url as positional argument. *Do not forget http:// (or https://).*

Example:

```
$ wight target-set http://api.wight.com
```

```
Wight target set to 'http://api.wight.com'. In order to login with wight, use 'wight login <email>'.
```

target-get

Gets the target wight is using currently.

Example:

```
$ wight target-get

Current Wight target is 'http://api.wight.com'. In order to login with wight, use 'wight login <emai
```

4.3.2 User Info Commands

In order to use the CLI, users need to authenticate with Wight, as well as be part of a team (more on that in the Team Commands section).

You can also obtain information about your currently logged-in user, like what teams you are part of or projects you can schedule tests for.

login

Logs in to Wight. Wight will keep your current authentication token in your home folder, meaning you won't have to login again until your token expiration.

If your user is not registered, Wight will ask you if you want to create a new account.

Example of registering a new account:

```
$ wight login test@gmail.com

Please enter the password to authenticate with (nothing will be displayed):
User does not exist. Do you wish to register? [y/n] y

User registered and authenticated.
```

Example of authenticating:

```
$ wight login test@gmail.com

Please enter the password to authenticate with (nothing will be displayed):

Authenticated.
```

change-password

This command allows the currently logged-in user to change his password.

Example:

```
$ wight change-password

Please enter your current password:

Please enter your new password:

Please enter your new password again:

Password changed successfully.
```

user-info

Shows user info, like what teams the user is part of or projects he’s allowed to schedule tests for.

Example:

```
$ wight user-info

User: bernardo@corp.globo.com
+----------+-------+
| team     | role  |
+----------+-------+
| timehome | owner |
+----------+-------+
```

4.3.3 Team Commands

Teams are the owners of projects and contain users. They are the unit of organization of everything in Wight.

The commands in this section are responsible for managing teams created in Wight, as well as assigning and removing users to teams.

team-create

Creates a team.

Example:

```
$ wight team-create myteam

Created ’myteam’ team in ’http://api.wight.com’ target.
```

team-update

Changes a team’s name.

Example:

```
$ wight team-update myteam newteam

Updated ’myteam’ team to ’newteam’ in ’http://api.wight.com’ target.
```

team-show

Show general information about the team, like it’s members and projects.

Example:

```
$ wight team-show newteam

newteam
-------

+---------------------+-------+
| user                | role  |
+---------------------+-------+
| heynemann@gmail.com | owner |
+---------------------+-------+

+--------------+----------------------------------------+----------------------+
| project name | repository                             | created by           |
+--------------+----------------------------------------+----------------------+
| myproject    | https://github.com/heynemann/wight.git | heynemann@gmail.com  |
+--------------+----------------------------------------+----------------------+
```

team-delete

Deletes a team.

WARNING: This operation cannot be undone and all the data (projects and tests) for the given team will be removed from Wight.

Example:

```
$ wight team-delete newteam

This operation will delete all projects and all tests of team ’newteam’.
You have to retype the team name to confirm deletion.

Team name:  newteam
Deleted ’newteam’ team, all its projects and tests in ’http://api.wight.com’ target.
```

team-adduser

Adds an user to a team. Being part of a team means that user gets to see the team test results, as well as schedule new tests.

Example:

```
$ wight team-adduser newteam test@gmail.com

User ’test@gmail.com’ added to Team ’newteam’.
```

team-removeuser

Removes an user from a team.

Example:

```
$ wight team-removeuser newteam test@gmail.com

User ’test@gmail.com’ removed from Team ’newteam’.
```

4.3.4 Project Commands

The commands in this section allow for management of the projects in a given team.

project-create

Creates a new project.

Example:

```
$ wight project-create myproject --team=myteam --repo=https://github.com/heynemann/wight.git

Created 'myproject' project in 'myteam' team at 'http://api.wight.com'.
```

project-update

Updates a project's information, like name or repository.

Example:

```
$ wight project-update --team=myteam --name=newname --repo=https://github.com/heynemann/wight.git wig

Updated 'newname' project in 'myteam' team at 'http://wight.target.com'.
```

project-delete

Deletes a project.

WARNING: This operation cannot be undone and all the tests for the given team will be removed from Wight.

Example:

```
$ wight project-delete --team=myteam --project=newname

This operation will delete the project 'newname' and all its tests.

Are you sure you want to delete project 'newname'? [y/n] y

Deleted 'newname' project and tests for team 'myteam' in 'http://wight.target.com' target.
```

4.3.5 Defaults Commands

In order to make it easier to use, Wight allows the user to specify a default team and default project.

When you have the defaults set you don't have to pass them in each command, as before.

default-set

Define default team and/or project to be used in subsequent commands.

Example:

```
$ wight default-set --team=myteam --project=myproject

Default team set to 'myteam'.

Default project set to 'myproject'.
```

default-get

Shows the defined default team and/or project.

Example:

```
$ wight default-get

Default team is 'myteam'.

Default project is 'myproject'.
```

4.3.6 Load Test Commands

list

Lists all the load tests for the projects that the currently logged-in user has access to.

The last three tests for each project get displayed.

Example:

```
$ wight list

Team: myteam ---- Project: newproject
+--------------------------------------+----------+---------------------+---------------------------
| uuid                                 |  status  |        since        |
+--------------------------------------+----------+---------------------+---------------------------
| c4dffdb6-1b67-42d8-8dac-c0534ee0065f | Finished |          -          | wight show c4dffdb6-1b67-42
| 925e8d32-b744-43a8-9236-bfd893b3a419 |  Failed  |          -          | wight show 925e8d32-b744-43
| ea7135dc-b63d-446b-82d1-cb2abaae2b6c |  Failed  |          -          | wight show ea7135dc-b63d-44
+--------------------------------------+----------+---------------------+---------------------------
```

show

Display a summary for one load test and all the results in it.

Example:

```
$ wight show c4dffdb6-1b67-42d8-8dac-c0534ee0065f

Load test: c4dffdb6-1b67-42d8-8dac-c0534ee0065f
Status: Finished
Based on commit: 3a68a2a05700649c15e15cf4c1d0b98962fb1768 by John Doe

+----------------+------------------+-------+------+--------+-------------------------------------
|     title      | concurrent users |  rps  | p95  | failed |
+----------------+------------------+-------+------+--------+-------------------------------------
| Login Page Tst |       100        | 264.0 | 0.31 |   0    | wight show-result 63fc4c0d-883f-444d-83
+----------------+------------------+-------+------+--------+-------------------------------------
                 rps means requests per second, p95 means the 95 percentile in seconds and failed mea
```

show-result

Show detailed information about one test result.

Example:

```
$ wight show-result 63fc4c0d-883f-444d-83c6-3d7cdffb5056

Load test: c4dffdb6-1b67-42d8-8dac-c0534ee0065f
Status: Finished
Web Report URL: http://web.wight.com/report/63fc4c0d-883f-444d-83c6-3d7cdffb5056

Bench Configuration
-------------------
Title: Login Page Tst                                   Description: Testing whether the login page wor
Module: test_login                                      Test: LoginScreenTest.test_login_screens
Cycles: [100, 200, 300]                                 Cycle Duration: 10s
Base URL: http://my.app.com/                            Test Date: 2013-08-16T17:44:16s

+-------+----------+---------+--------+---------+---------+-------+-------+---------+
| users | requests | error % |  rps   | minimum | average |  p90  |  p95  | maximum |
+-------+----------+---------+--------+---------+---------+-------+-------+---------+
|  100  |   2640   |  0.00%  | 264.00 |  0.03s  |  0.12s  | 0.26s | 0.31s |  1.17s  |
|  200  |   2833   |  0.00%  | 283.30 |  0.08s  |  0.44s  | 0.70s | 0.88s |  3.67s  |
|  300  |   2812   |  0.00%  | 281.20 |  0.08s  |  0.52s  | 0.78s | 1.21s |  6.92s  |
+-------+----------+---------+--------+---------+---------+-------+-------+---------+
rps means requests per second and average, p95 and maximum are all response time in seconds
```

schedule

Schedules a new load test.

Example:

```
$ wight schedule http://site.tobehitten.com --team my-team --project my-project

Scheduled a new load test for project ’my-project’ in team ’my-team’ at ’http://wight.targetset.com’
```

Variations

Schedule a simple test in a URL for a project with no funkload tests:

```
$ wight schedule http://site.tobehitten.com --team my-team --project my-project --simple

Scheduled a new load test for project ’my-project’ in team ’my-team’ at ’http://wight.targetset.com’
```

Schedule a test where the bench folder are in a specific git branch:

```
$ wight schedule http://site.tobehitten.com --team my-team --project my-project --branch test-branch

Scheduled a new load test for project ’my-project’ (branch ’test-branch’) in team ’my-team’ at ’http
```

If you had set default team and/or project, you can omit these parameters in schedule command:

```
$ wight schedule http://site.tobehitten.com

Scheduled a new load test for project ’my-project’ in team ’my-team’ at ’http://wight.targetset.com’
```

4.4 API

TBW.

4.5 Provisioning

TBW.

4.6 Architecture

TBW.

4.7 Release Notes

TBW.

4.8 Contributors

TBW.

4.9 Licensing

Wight is licensed under the MIT License: